

CLAIMS

- 1 1. A method for generating test cases, comprising:
2 providing rule-based generation of test cases from an abstract representation
3 that includes application states, external interaction sequences and input data of test
4 cases from data stores;
5 validating generated test cases; and
6 converting the test cases to test scripts.
- 1 2. The method of claim 1, wherein a data store is a relational database
2 management system.
- 1 3. The method of claim 1, wherein a data store is an XML database
2 management system.
- 1 4. The method of claim 1, wherein a data store is a file system.
- 1 5. The method of claim 1, wherein an application state represents a
2 runtime snapshot of application under test which defines the context of external
3 interaction.
- 1 6. The method of claim 5, wherein the application state includes a set of
2 application objects, its attributes and attribute values.
- 1 7. The method of claim 5, wherein the application states corresponding to
2 a test case are arranged in a hierarchical manner.
- 1 8. The method of claim 1, wherein the external interaction sequences
2 represent events invoked by external agents on the application objects.
- 1 9 The method of claim 8, wherein the external agents are human agents
2 or other software agents.
- 1 10. The method of claim 8, wherein the interaction sequencing includes
2 flow control structures for capturing sequential, concurrent, looping and conditional
3 interactions.

- 1 11. The method of claim 1, wherein the validation of generated test cases
2 includes internal and external validation.
- 1 12. The method of claim 11, wherein the internal validation ensures that
2 the components of the test case definition, external interaction sequences and input
3 data are consistent with each other and with an application object model.
- 1 13. The method of claim 12, wherein an application object model is a
2 metadata representation for modeling application under test.
- 1 14. The method of claim 13, wherein the metadata representation includes
2 object type definitions for application objects.
- 1 15. The method of claim 13, wherein the metadata representation includes
2 attribute definitions for each application object type.
- 1 16. The method of claim 13, wherein the metadata representation includes
2 definition of methods and events that are supported by each application object type.
- 1 17. The method of claim 13, wherein the metadata representation includes
2 definition of effects of events on an application state.
- 1 18. The method of claim 14, wherein application object type definitions
2 include additional categorization of each application object types into hierarchical,
3 container and simple types.
- 1 19. The method of claim 18, wherein the hierarchical object types are
2 associated with an application state of its own, wherein application object types that
3 can contain instances of other objects are termed container types.
- 1 20. The method of claim 19, wherein the state associated with a
2 hierarchical application object type is a modal application state or a nonmodal
3 application state.
- 1 21. The method of claim 20, wherein a modal application state restricts
2 possible interactions to application object instances available within the current
3 application state.

1 22. The method of claim 17, wherein the effects of events on an application
2 state capture one or more consequences of the event to the application state.

1 23. The method of claim 22, wherein a consequence of an event is
2 selected from, creation of a new object instance of a given type, deletion of an object
3 instance of a given type, modification of attributes of an existing object instance and
4 selection of an instance of an object type.

1 24. The method of claim 23, wherein creation of a new instance of an
2 object of type that is hierarchical results in creation of a new application state.

1 25. The method of claim 23, wherein selection of an object instance of type
2 that is hierarchical results in selection of the application state associated with that
3 object instance.

1 26. The method of claim 11, wherein the external validation validates the
2 generated test case against the application metadata repository.

1 27. The method of claim 26, wherein the application metadata repository
2 contains definition of application objects and nature of their interactions within the
3 application under test.

1 28. The method of claim 26, wherein the external validation serves as a
2 static verification test for the test cases.

1 29. The method of claim 26, wherein the external validation increases
2 productivity by pointing out invalid test cases.

1 30. The method of claim 26, wherein the external validation increases
2 productivity by pointing out inconsistencies in statically verifiable application
3 behaviors.

1 31. The method of claim 1, wherein the test scripts are test cases
2 represented in a scripting language.

1 32. The method of claim 31, wherein the scripting languages can be typed
2 or untyped programming languages used for recording or authoring test cases.

1 33. The method of claim 1, further comprising:
2 providing rules for selection of components of test case definition, external
3 interaction sequences and input data; rules for data driven test case generation.

1 34. The method of claim 33, wherein the selection rules are specified using
2 query languages.

1 35. The method of claim 34, wherein the query language is Structured
2 Query Language (SQL).

1 36. The method of claim 34, wherein the query language is XML Query
2 (XQuery).

1 37. The method of claim 34, wherein the query language is Application
2 Programming Interface (API) called from code written in a programming language.

1 38. The method of claim 34, wherein the use of query languages allows
2 test cases to be generated from live customer data.

1 39. The method of claim 33, wherein the data driven test case generation
2 involves composing the test case as dictated by the input data.

1 40. The method of claim 39, wherein the availability of multiple datasets for
2 the input data will result in generation of multiple test cases or external interaction
3 sequences repeated within a loop control structure for each dataset.

1 41. The method of claim 39, wherein the availability of multiple datasets for
2 a portion of the input data will result in the interaction sequences corresponding to
3 this portion of input data repeated within a loop control structure.

1 42. The method of claim 39, wherein each element of input data can be
2 flagged valid or invalid.

1 43. The method of claim 42, wherein the presence of validity flag in the
2 input data that is different from the one corresponding the input data when the test
3 cases was recorded or authored, results in the generator including appropriate
4 interaction sequences for exception handling.

1 44. The method of claim 1, further comprising:
2 converting test case from internal representation to a scripting language
3 through language mapping.

1 45. The method of claim 44, wherein the mapping is used to map external
2 interactions captured as events on application object to appropriate statements in the
3 scripting language.

1 46. The method of claim 44, wherein more than one language mappings
2 are provided at the same time.

1 47. The method of claim 44, wherein the generated test case are
2 converted to more than one scripting language at the same time.

1 48. The method of claim 47, wherein generating test cases in multiple
2 scripting language allows generation of test scripts for multiple test execution
3 environments.

1 49. A computer system, comprising:
2 a processor;
3 a memory coupled to the processor, the memory storing rule-based
4 generation of test cases from an abstract representation that includes application
5 states, external interaction sequences and input data of test cases from data stores
6 to produce test cases;
7 logic that validates the test cases; and
8 logic for converting the test cases to test scripts.

1 50. The system of claim 49, wherein the logic that validates the test cases
2 provides that components of a test case definition, external interaction sequences
3 and input data are consistent with each other and with an application object model.

1 51. The system of claim 49, wherein the logic that validates the test cases
2 is external validation logic.

1 52. The system of claim 51, wherein the external validation logic includes
2 validating a generated test case against an application metadata repository.

1 53. The system of claim 49, further comprising:
2 logic for providing rules for selection of components of test case definition,
3 external interaction sequences and input data; wherein and the rules are data driven
4 test case generation.

1 54. The system of claim 49, further comprising:
2 logic for providing data driven test case generation.

1 55. The system of claim 54, wherein the logic for providing data driven test case
2 generation includes composing the test case as dictated by the input data..